Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2010, Article ID 808797, 14 pages
doi:10.1155/2010/808797

Research Article

Distributed KDC-Based Random Pairwise Key Establishment in Wireless Sensor Networks

Zhong Su,[1] Yixin Jiang,[1] Fengyuan Ren,[1] Chuang Lin,[1] and Xiaowen Chu[2]

[1] *Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China*
[2] *Department of Computer Science, Hong Kong Baptist University, Hong Kong*

Correspondence should be addressed to Xiaowen Chu, chxw@comp.hkbu.edu.hk

Received 2 March 2010; Accepted 1 July 2010

Academic Editor: Jiangchuan Liu

Key management is a core mechanism to provide secure and reliable communications in wireless sensor networks (WSNs). In large-scale WSNs, due to the resource constraint on sensor nodes, it is still an extremely challenging task to achieve good performance in terms of high network connectivity and strong resilience against sensor nodes capture with low overheads. To address this issue, in this paper we propose a novel random pairwise key establishment scheme, called RPKE. In RPKE, sensor nodes differentiate their roles as either auxiliary nodes or ordinary nodes prior to network deployment. The auxiliary nodes act as distributed key distribution center (KDC), and neighboring ordinary nodes can establish pairwise key with the help of the distributed KDC. Theoretical analysis and simulation evaluation demonstrate that RPKE performs well in terms of network connectivity and resilience at the cost of low computation/communication/storage overheads, compared to the existing counterparts.

1. Introduction

Wireless Sensor Networks (WSNs) provide many promising applications such as, pollution sensing, environment, and traffic monitoring [1]. Security is a critical issue especially when WSN is deployed in hostile environment where sensor nodes may be exposed to a variety of malicious attacks. One of the fundamental problems in WSN is how to bootstrap secure communications, that is, how to establish pairwise keys between sensor nodes in order to offer data confidentiality and data integrity.

Large-scale WSNs consist of a large number of sensor nodes [2, 3]. Usually, sensor nodes have limited capacity in terms of computation power, communication range, and storage space. For example, the MICA2 mote has an 8-bit 7.3828-MHz Atmega 128 L processor with only 4-Kbyte SRAM and 128-Kbyte ROM [4]. Hence, classical asymmetric cryptography such as, RSA [5] or centralized key agreement scheme [6] is unsuitable for WSN due to limited resources.

Recently, symmetric key predistribution schemes [7–9] have been proposed to achieve secure communications in WSNs. In key predistribution schemes, sensor nodes preload some keys or keying material prior to network deployment and establish pairwise keys by exchanging partial keying information after network deployment. A trivial solution is to distribute a shared master key to all sensor nodes, so each pair of nodes can establish secure communication link with less storage. However, this trivial scheme offers the worst resilience because the adversary can compromise all the communication links even though he only compromises a single sensor node. Another naïve solution is to distribute unique pairwise keys for all pair of sensor nodes; the adversary cannot compromise the communication links between two noncompromised sensor nodes no matter how many nodes have been compromised. However, each node must store N-1 keys where N is the network size, so this naïve solution is not scalable for large-scale WSNs due to the storage constraint in sensor nodes. Single key distributed center-(KDC-) based key predistribution scheme [10] can efficiently reduce the storage cost for sensor nodes, but it incurs large communication cost for sensor nodes and suffers from a single point of failure.

Random key predistribution schemes [7, 11–19] have recently attracted much attention, in which sensor nodes

randomly pick a part of keys out of a large key pool prior to network deployment. After network deployment, neighboring nodes share common keys with a certain probability and can establish pairwise keys using these common keys. Hence, the random key predistribution schemes are considered as the most practical ones in WSNs due to their distributed nature and simplicity.

However, due to the random predistribution, the preloaded the number of keys in each node will increase linearly with the total number of nodes if the desirable network connectivity probability is required, which will incur a high storage burden in large-scale WSNs. For security, the nodes are expected to be preloaded with a small number of keys. The smaller number of preloaded keys, the less number of keys will be acquired by the adversary when a node is compromised. To achieve high performance, some efficient schemes have been presented to establish pairwise key, by employing multiple polynomials [13], location information [14, 15], deployment knowledge [16], multiple key spaces [17], or heterogeneity [18]. However, such schemes either incur high computation burden [13, 17] or make assumptions that may not be always available in typical WSNs [14–17]. Therefore, it is still an extremely challenging task to achieve high network connectivity, strong resilience against node capture, and low storage/computation/communication overheads.

Motivated by this, in this paper we propose a novel Random Pairwise Keys Establishment (RPKE) scheme for WSNs, in which nodes differentiate their roles as auxiliary or ordinary nodes prior to network deployment. After network deployment, auxiliary nodes serve as distributed KDCs to help pairwise key establishment between ordinary nodes. Two key pools, namely, initial key pool and root key pool, are constructed for auxiliary nodes, and ordinary nodes respectively. With the help of auxiliary nodes which preload a large number of initial keys, ordinary nodes only preload a small fraction of root keys and can establish pairwise key with high probability while keeping stronger resilience against node compromise.

The main advantages of RPKE include the following. (1) Efficiency: the RPKE scheme is very suitable for large-scale WSNs, where the distributed KDCs can efficiently distribute the keying material to neighboring ordinary nodes during pairwise key establishment, thus the storage/communication/computation overheads for the ordinary nodes are significantly reduced. (2) Robustness: the RPKE scheme is very robust against node compromise. By constructing two types of key pool for two kinds of sensor nodes, respectively, the secret keys (initial keys and root keys) are stored separately in auxiliary nodes and ordinary nodes. Thus, it is difficult for the adversary to acquire the pairwise key between noncompromised ordinary nodes by capturing arbitrarily a part of sensor nodes. (3) Flexibility: according to the different application scenarios, the security parameters in RPKE can be conveniently tuned to achieve excellent network connectivity and high security strength with very low overhead requirement in ordinary nodes.

The rest of this paper is organized as follows. In Section 2, we review the related works. The background and some preliminaries related to the proposed scheme are given in Section 3. In Section 4, the proposed RPKE scheme is introduced in detail. The network performance and security analysis are, respectively, presented in Sections 5 and 6, followed by conclusions in Section 7.

2. Related Works

In literatures, various key predistribution schemes have been proposed for securing WSNs.

SNEP [10] is a *single KDC-based key predistribution scheme*, where each node only preloads a symmetric key shared between itself and the base station, which acts as a single KDC. If two sensor nodes want to establish pairwise key, they must communicate with the base station, and then the base station assigns the pairwise key for them. Clearly, in the large-scale WSNs, SNEP will incur high communication burden for those sensor nodes near to the base station. Furthermore, the single point of failure will break the security of the entire network.

Eschenauer and Gilgor [7] firstly propose *random key predistribution scheme*, which is referred as basic scheme in this paper. In basic scheme, prior to network deployment, each sensor node preloads a key ring with a randomly chosen subset of keys from a large key pool without replacement, and two neighboring nodes have some probability p of successfully completing key establishment. Due to the random key predistribution, it is probable that a shared key may not be available, necessitating the intermediary nodes with common keys between the two sensor nodes to establish pairwise key for them. The q-composite key predistribution scheme [11] is a modified version of the basic scheme, differing only in the fact that multiple keys are used to establish pairwise key instead of just one. By increasing the amount of key overlap required for key establishment, this scheme increases the resilience against node compromise. However, the basic scheme and the q-composite scheme cannot achieve good performance in the large-scale WSNs. The number of compromised communication links between noncompromised neighboring sensor nodes will dramatically increase with the number of compromised sensor nodes. Recently, Blackshear and Verma [12] propose a randomizing LEAP+ key distribution scheme to resist the node compromise attack which is vulnerable in basic scheme.

To enhance the security, Liu and Ning [13] propose a multiple polynomial-based random key predistribution scheme in which each node randomly preloads a subset of polynomial shares, two neighboring nodes can establish pairwise key if they have the polynomial share on the same bivariate polynomial. Due to the λ-secure property of polynomial (i.e., the polynomial remains secure if no more than λ polynomial shares are compromised), the scheme has good resilience; however, the required $O(\lambda)$ modular multiplications incur large computation overhead. Similarly, in the multiple key space-based scheme [14] with λ-secure property, the computation burden makes it not scalable for large-scale WSNs.

To reduce the storage requirement in nodes, based on expected locations of the nodes, Liu and Ning [15] present

a location-aware random key predistribution scheme. Any two nodes would share a common pairwise key if both of them expect to appear in each other's signal range with a high probability. Huang et al. [16] propose a grid-group deployment scheme to improve resilience against selective node capture and node fabrication attack. Du et al. [17] further propose a random key predistribution scheme by exploiting the node deployment knowledge such that the probability to find a common secret key between any two neighboring nodes can be maximized while other performance metrics are not degraded. Although such schemes achieve good performance in terms of connectivity and resilience, the pre-determined location information or the deployment knowledge, however, are not always available in typical WSNs.

Traynor et al. [18] proposed an unbalanced random key predistribution scheme for Heterogeneous WSNs, where there are a large number of the less capable nodes (L1) and a small number of the more capable nodes (L2). Fewer keys are preloaded in L1 nodes while more keys are preloaded in L2 nodes, and L1 nodes and L2 nodes can achieve secure connection in different scenarios. In this scheme, L2 nodes are the bottleneck of network connectivity and resilience.

Vu et al. [19] figure out that most random key predistribution schemes are vulnerable to the node capture attack, and then propose virtual key ring technique to strengthen the resilience by reducing the preloaded keying material while maintaining secure connectivity of the network.

iPAK [20], LKE [21], and SBK [22] are the In-Situ key establishment schemes. In these schemes, nodes differentiate their roles as service sensors and worker sensors. Service sensors are used to disseminate keying information to the work sensors in the vicinity after network deployment. One benefit of these schemes is that all the work sensors need not preload any keying information and can directly establish pairwise key with neighboring worker sensors, which can save storage space greatly; however, the major drawback of these schemes is that it must use Rabin's cryptosystem to establish secure channel between the work sensors and service sensors. Thus, such schemes may incur higher communication/computation cost and have worse topology adaptability.

3. Preliminaries

3.1. System Model. We consider a WSN consisting of two types of sensor nodes, *ordinary nodes* and *auxiliary nodes*. Ordinary nodes are in charge of normal network operation, whereas auxiliary nodes are to offer keying material to help the pairwise key establishment for ordinary nodes and do not participate in other further network operation.

The number of auxiliary nodes is much smaller than that of the ordinary nodes. Moreover, due to the nature of random deployment, there is not any deployment or neighbor information available for all the sensor nodes prior to network deployment.

In our consideration, all the sensor nodes are not assumed to be equipped with tamper-resistant hardware due to resource constraints and can directly communicate only with a limited number of other sensor nodes located in the communication range.

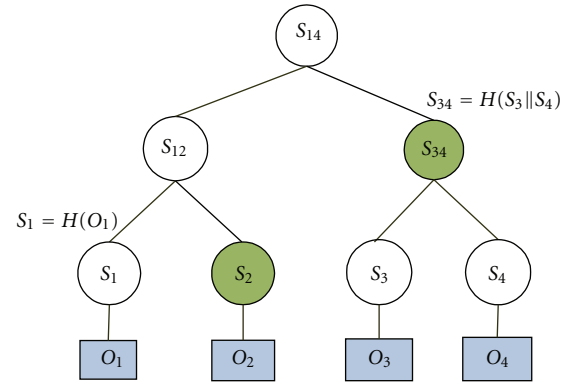


Figure 1: Merkle hash tree with four objects: O_1, O_2, O_3, and O_4.

3.2. Threat Model. We assume that the adversary has more powerful resources in terms of energy, computation, and communication capacity than sensor nodes. The adversary can compromise a fraction of sensor nodes chosen arbitrarily in WSN. Moreover, the adversary can expose all the secret information within the compromised sensor nodes. However, all the sensor nodes must be designed to survive at least a short interval when captured by an adversary.

In this work, the goal of the adversary is the exposure of the pairwise key between two noncompromised ordinary nodes. If a pairwise key is acquired by the adversary, the data confidentiality in the link will no longer be secure. To achieve the goal, the adversary can either overhear the transmitted message through the radio communication channel or physically capture any sensor nodes.

3.3. Merkle Hash Tree. Merkle hash tree [23] is a complete binary tree which is usually used to offer identity authentication. We use Merkle hash trees to verify initial key during pairwise keys establishment. Specially, any auxiliary nodes cannot forge bogus initial keys to cheat ordinary nodes, and the ordinary nodes only accept those initial keys from the real key groups.

In a Merkle hash tree, the leaf nodes are the hash values of the authentic objects, and the interior nodes are the hash values of the concatenation of its two children nodes. Each of the leaf nodes has its authentication path, referred to as *IDCert*, which consists of the sibling nodes of the nodes on the path from the leaf node to the root of tree (excluding the root). To verify the authenticity of an object, one could compute a value using the corresponding *IDCert* and compare the computed result with the public root value.

Figure 1 depicts an example of Merkle hash tree where there are four objects O_1, O_2, O_3, and O_4. The values of leaf nodes are $S_i = H(O_i)$ $(i = 1, 2, 3, 4)$, where H is a secure one-way hash function [24]. The interior node S_{34} is calculated as $S_{34} = H(S_3 \| S_4)$, where "$\|$" denotes the concatenation operation.

When one wishes to verify authenticity of object O_1, he can do so by using the hash value of O_1 along with O_1's authentication path $IDCert_{O_1} = \{S_2, S_{34}\}$. With these values, the one who knows the root value S_{14} can verify the authenticity of O_1 by checking if $S_{14} = H(H(H(O_1)\|S_2)\|S_{34})$.

4. The Proposed Scheme

In this section, we propose the RPKE, a random pairwise key establishment scheme for WSNs using auxiliary nodes. RPKE is divided into three phases: (1) *predeployment*, which specifies how to preload keying material to each ordinary node and auxiliary node; (2) *derived keys acquisition*, which specifies how to construct the derived key shared between two neighboring ordinary nodes with help of the common auxiliary node(s), and (3) *pairwise key establishment*, which specifies how to establish a pairwise key using the derived keys shared between two neighboring ordinary nodes.

4.1. Predployment Phase. During the predeployment phase, a trusted offline server first generates two key pools, initial key pool and root key pool; then the auxiliary nodes and the ordinary nodes are preloaded the key materials, respectively.

4.1.1. Generating Key Pool. The initial key pool consists of L key groups and each key group consists of M initial keys. Hence, the initial key pool contains $P = L * M$ initial keys, where L and M are system parameters based on the network connectivity and security requirements. The jth key in the ith key group of the initial key pool is denoted as $k_{i,j}$ ($i \in [1, L]$, $j \in [1, M]$).

Furthermore, every key group has an identifier, that is, GID_i is the identifier of the ith key group. Note that the key group identifier is a private hash value.

Once the initial key pool is generated, the trusted server constructs L Merkle hash trees, described in Section 3.3. Each Merkle hash tree corresponds to a key group. The leaves of Merkle hash tree are generated by hashing initial keys in the key group (For convenience, assume M is the power of 2). Hence, there are M leaves in each of Merkle hash tree and L root values for total Merkle hash trees.

The root key in the root key pool is generated by hashing concatenation of the root value and its associated group identifier, that is, $R_i = H(r_i\|GID_i)$ (r_i is the root value of the ith Merkle hash tree, where $i \in [1, L]$). Hence, the root key pool can be denoted as $\{R_1, R_2, \ldots, R_L\}$.

4.1.2. Preloading Keying Material. For each ordinary node, it needs to pick the following secret information:

(1) q_n ($q_n \ll L$) root keys out of the root key pool without replacement to establish its key ring,

(2) q_n associated with group identifiers, that is, if the root key R_i is preloaded, the group identifier GID_i must also be preloaded.

For each auxiliary node, it needs to pick

(1) q_a initial keys from the initial key pool without replacement, where $q_a \ll L * M$. Note that we do not have any limitation on the number of initial keys selected from the same key group. As a result, some of these keys may come from the same key group,

(2) the associated *IDCert* of every picked initial key,

(3) a hash image of the root key $H(R_i)$ if there is at least one initial key of ith key group to be picked. For example, if the auxiliary node picks the initial key $k_{2,3}$, it will also preload $H(R_2)$.

After being preloaded with corresponding keying material, all nodes, including ordinary nodes and auxiliary nodes, are randomly deployed in a sensed area.

4.2. Derived Key Acquisition Phase. The derived key acquisition phase occurs after the network deployment. Initially, each auxiliary node broadcasts a *Hello* message, announcing its existence to ordinary nodes within h-hop away. As a result, all the ordinary nodes know which auxiliary node is neighbor. Note that the hop count h is a designed system parameter which determines the number of common auxiliary nodes for two neighboring ordinary nodes. We will discuss how the hop count along with network degree and radio range affects the number of common auxiliary nodes in Section 5.

When an ordinary node U wants to establish pairwise key with its neighboring ordinary node V, U will send all of its root key identifiers and its auxiliary node identifiers to V. V determines that it shares one of the root keys associated with U, and responds to U a challenge/response. U and V exchange the messages 1 and 2 as shown in Figure 2:

$$\begin{aligned} &① \; U \longrightarrow V : ID_U, N_U, \{ID_{R_i}\}_{i=A_1,\ldots,A_{q_n}}, \\ &\qquad \left\{ID_{AN_j}\right\}_{j=A_1,\ldots,A_s}, \\ &② \; V \longrightarrow U : ID_U, ID_V, N_U, ID_{r_1}, \\ &\qquad \left\{\{ID_{R_i}\}_{i=r_1,\ldots,r_t}, \left\{ID_{AN_j}\right\}_{j=B_1,\ldots B_g}, ID_V, N_U\right\}_{\langle R_{r_1}\rangle}. \end{aligned} \tag{1}$$

The nonce N_U is used to defense the replay attack. In this case, both U and V know that they share t root keys and they have g common auxiliary nodes.

If U and V share at least one root key, they may generate derived key(s) with the help of the common auxiliary nodes. For this, U and V transmit their shared root keys identifiers to their common auxiliary node(s) as follows:

$$\begin{aligned} &③ \; U \longrightarrow AN : ID_U, ID_V, N_U, \{ID_{R_i}\}_{i=r_1,\ldots,r_t}, \\ &④ \; V \longrightarrow AN : ID_V, ID_U, N_V, \{ID_{R_i}\}_{i=r_1,\ldots,r_t}. \end{aligned} \tag{2}$$

Once receiving the transmitted messages from U and V, the auxiliary node AN will act as a KDC and send one or more reply packets to U and V. If ID_{R_i} is the common key identifier and AN has preloaded one or more initial keys from the ith key group, AN will send reply messages to U

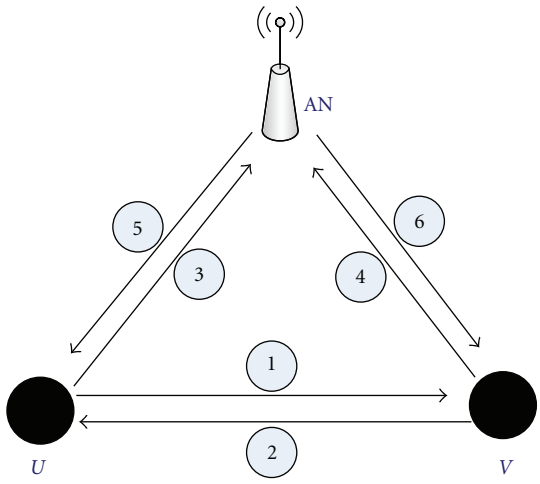


Figure 2: Interaction among ordinary nodes U, V, and auxiliary node AN.

and V. The reply message contains all the preloaded initial keys of ith key group and their associated *IDCerts*

$$\begin{aligned}
&⑤\ AN \longrightarrow U : ID_{AN}, ID_V, N_U, ID_{R_i},\\
&\left\{k_{i,j_1}, IDCert_{i,j_1}, \ldots, k_{i,j_c}, IDCert_{i,j_c}, ID_{AN}, ID_V, N_U\right\}_{\langle H(R_i)\rangle},\\
&⑥\ AN \longrightarrow V : ID_{AN}, ID_U, N_V, ID_{R_i},\\
&\left\{k_{i,j_1}, IDCert_{i,j_1}, \ldots, k_{i,j_c}, IDCert_{i,j_c}, ID_{AN}, ID_U, N_V\right\}_{\langle H(R_i)\rangle}.
\end{aligned} \tag{3}$$

If there are more than one common root key identifier, *AN* may send more than one reply message if it preloads at least one initial key from the corresponding key groups.

When U receives the reply message from the common auxiliary nodes *AN*, it first authenticates the initial keys using their *IDCert*s, the associated group identifier, and the associated root key. If the $k_{i,j}$ has been authenticated, U will generate a derived key as follow:

$$K_{i,j,AN} = H\left(R_i \| k_{i,j} \| ID_{AN}\right). \tag{4}$$

V also can generate the similar derived keys in this phase.

4.3. Pairwise Key Establishment Phase. Based on the identifiers knowledge of common neighboring auxiliary nodes and common root keys, U certainly knows which derived key it can share with V. Assume that U and V have in common derived keys $\{K_1, K_2, \ldots, K_m\}$ with m above the threshold q, U and V can establish pairwise key as follows:

$$K_{UV} = K_1 \oplus K_2 \oplus \cdots \oplus K_m, \tag{5}$$

where "$\oplus$" denotes the bitwise exclusive OR operation. The ordinary nodes will erase the derived keys when the pairwise key has been generated.

Two neighboring ordinary nodes U and V may not share root keys due to the randomness in root keys predistribution. If so, they cannot directly establish pairwise key with help of their common auxiliary nodes. In such case, we adopt the approach similar to that given in [17] to establish pairwise key for U and V. That is, assume there is a key path $\{U, V_1, V_2, \ldots, V_j, V\}$ in which each pair of ordinary nodes, $(U, V_1), (V_1, V_2), \ldots, (V_j, V)$, has established the secure link. U first generates a random key K and sends the key to V_1 using their secure link, V_1 sends the key to V_2 using the secure link between V_1 and V_2, and so on until V receives the key from V_j, U and V will use the key K as their pairwise key. The length of key path is the number of intermediate nodes, that is, $\{V_1, V_2, \ldots, V_j\}$.

5. Performance Evaluations

In the following section, we discuss the performance metric for the proposed RPKE scheme and compare it with other classical random key predistribution schemes [7, 11].

5.1. Network Connectivity. Since the ordinary nodes are in charge of normal network operations, the "*network connectivity probability*" is defined as the probability of establishing secure communication link between two neighboring ordinary nodes. As shown in (5), if two neighboring ordinary nodes share enough number of derived keys, they can establish pairwise key, that is, establish a secure communication link.

Two neighboring ordinary nodes U and V must share derived keys if the following conditions hold:

(1) U and V share at least one root key;

(2) at least one of their common auxiliary nodes picks one or more initial keys from the key groups whose root keys are shared by U and V.

Now we discuss how to satisfy the two above-mentioned conditions in order to establish pairwise key between two neighboring ordinary nodes.

Let m_i be the number of common derived keys between U and V through the common auxiliary node AN_i. The total number of common derived keys m between U and V through all their g common auxiliary nodes thus is $m = m_1 + m_2 + \cdots + m_g$. Then we can give the following conclusions.

Lemma 1. *Assuming that two neighboring ordinary nodes U and V have only a common auxiliary node, if U shares exactly i root keys with V, the common derived keys shared between them must be no more than $i * M$.*

Proof. Each root key can authenticate at most M initial keys, since the common derived keys must come from key groups which root keys are shared between U and V, if U and V share i root keys and the common auxiliary node picks all the $i * M$ initial keys of the total i key groups, these keys will be authenticated by U and V, then they will share $i * M$ common derived keys. Otherwise, if any one of the initial keys among these $i * M$ keys is not picked by the common auxiliary node, the common derived keys shared between U and V must be no more than $i * M$. □

Theorem 2. *Assume that neighboring ordinary nodes U and V have g common auxiliary nodes, and the number of common derived keys shared between U and V through each of g common auxiliary nodes is $m_1, m_2, \ldots, m_g$, respectively (for all $i, m_i \ll q_n$); then the number of root keys shared between U and V must be at least $\lceil \max(m_1, m_2, \ldots, m_g)/M \rceil$.*

Proof. If U and V share i root keys, according to Lemma 1, the common derived keys shared between them must range from 0 to $i * M$. If they can generate m_i common derived keys through the common auxiliary node AN_i, they must share at least $\lceil m_i/M \rceil$ root keys. For all the m_i $(i = 1, 2, \ldots, g)$, if U and V share enough number of root keys to ensure generating the maximum common derived keys amongst $m_1, m_2, \ldots, m_g$ through one of common auxiliary nodes, they are able to generate other common derived keys through other common auxiliary nodes. Therefore, the number of shared root keys must be at least $\lceil \max(m_1, m_2, \ldots, m_g)/M \rceil$. □

Assume that U shares i root keys with V and there is only one common auxiliary node between U and V. Obviously, U and V can generate at most $i * M$ common derived keys. To ensure U and V share m common derived keys, there are $\binom{i \times M}{m}$ ways to let the common auxiliary node select m different initial keys from these i key groups. Similarly, there are $\binom{(L-i) \times M}{q_a - m}$ ways to let the common auxiliary node randomly select $(q_a - m)$ different initial keys from the remaining $(L - i)$ key groups. Hence, the total number of ways that the common auxiliary node selects initial keys from initial key pool should be

$$\Omega(i, m) = \binom{i \times M}{m} \binom{(L - i) \times M}{q_a - m}. \tag{6}$$

When two neighboring ordinary nodes U and V have g common auxiliary nodes, the total number of ways that U and V share m derived keys can be calculated as follows:

First, U and V have $\binom{L}{q_n}$ different ways to randomly select q_n root keys from root key pool.

Second, the shared derived keys, that is, $m_1, m_2, \ldots, m_g$ between U and V, must range from 0 to m and $m = m_1 + m_2 + \cdots + m_g$. Hence there are $\sum_{m_1+m_2+\cdots+m_g=m}$ ways to let every common auxiliary node provide shared initial keys. According to Theorem 2, U must share at least $\lceil \max(m_1, m_2, \ldots, m_g)/M \rceil$ root keys with V. Once $m_1, m_2, \ldots, m_g$ are fixed, the number of shared root keys will range from $\lceil \max(m_1, m_2, \ldots m_g)/M \rceil$ to q_n. Hence, the total number of ways by which V randomly selects root keys from the root key pool is given by $\sum_{i=\lceil \max(m_1,m_2,\ldots,m_g)/M \rceil}^{q_n} \binom{q_n}{i} \binom{L-q_n}{q_n-i}$. The total number of ways that g common auxiliary nodes randomly select initial keys is $\Omega(i, m_1)\Omega(i, m_2) \cdots \Omega(i, m_g)$, where $\Omega(i, m_j)$ for $j = 1, 2, \ldots,$ g is defined in (6).

Hence, the probability that two neighboring ordinary nodes share m $(m \geqslant 0)$ derived keys when they have g $(g \geqslant 1)$ common auxiliary nodes can be calculated as follows:

$$p(m) = \frac{\sum_{m_1+m_2+\cdots+m_g=m} \sum_{i=\lceil \max(m_1,m_2,\ldots,m_g)/M \rceil}^{q_n} \binom{q_n}{i} \binom{L-q_n}{q_n-i} \Omega(i, m_1) \cdots \Omega\left(i, m_g\right)}{\binom{L}{q_n} \binom{L \times M}{q_a}^g}. \tag{7}$$

Let p_{connect} be the probability of two neighboring ordinary nodes sharing sufficient derived keys to establish pairwise key. Obviously, $p_{\text{connect}} = 1 - \text{Prob.}\{\text{two neighboring ordinary nodes share insufficient derived keys to establish pairwise key}\}$. Hence, we have

$$\begin{aligned} p_{\text{connect}} &= 1 - (p(0) + p(1) + \cdots + p(q-1)) \\ &= 1 - \frac{\sum_{m_1=0}^{q-1} \sum_{m_2=0}^{q-m_1-1} \cdots \sum_{m_g=0}^{q-m_1-\cdots-m_{g-1}-1} \sum_{i=\lceil \max(m_1,m_2,\ldots,m_g)/M \rceil}^{q_n} \binom{q_n}{i} \binom{L-q_n}{q_n-i} \Omega(i, m_1) \cdots \Omega\left(i, m_g\right)}{\binom{L}{q_n} \binom{L \times M}{q_a}^g}. \end{aligned} \tag{8}$$

From (8), we can conclude that the system parameters, such as, the number of common auxiliary nodes, the size of the initial key group, the number of initial key groups, and may influence the network connectivity performance.

5.2. Impact of the System Parameters. One of design goals for the RPKE is to achieve high network connectivity while ordinary nodes only need to preload a few keying materials. In this subsection, we determine the number of required root keys to achieve targeted network connectivity using the above equations. We discuss these system parameters through both theoretical analysis and simulation studies. Note that the following analysis is required to achieve 99.99% network connectivity probability.

5.2.1. Number of Common Auxiliary Nodes. The number of derived keys for ordinary nodes relies on the number of their neighboring auxiliary nodes. Hence, the larger the number of common auxiliary nodes for the two neighboring ordinary nodes, the higher the probability that they share common derived keys. Figure 3 shows the number of root keys required in every ordinary node versus the number of

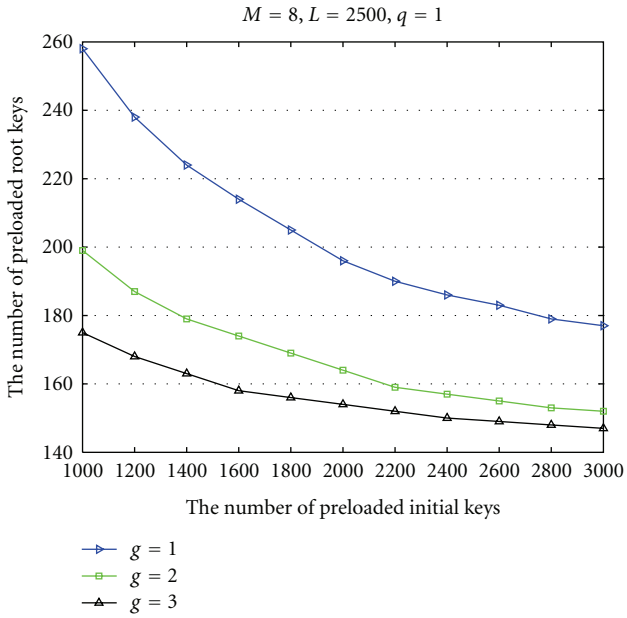


Figure 3: The number of preloaded root keys versus the number of preloaded initial keys varying with the number of common auxiliary nodes.

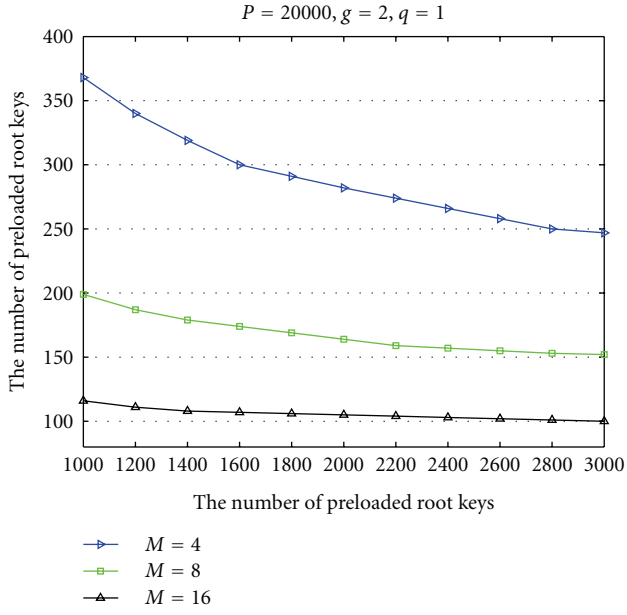


Figure 4: The number of preloaded initial keys versus the number of preloaded root keys varying with different size of key group.

initial keys preloaded in auxiliary nodes varying with the number of common auxiliary nodes.

This chart offers the results as expected in the aforesaid observation, that is, the storage overhead in ordinary nodes reduces with the increase of the number of common auxiliary nodes. In addition, it also shows an interesting phenomenon, that is, increasing the number of common auxiliary nodes does not result in a remarkable decrease of the number of preloaded root keys when the auxiliary node preloads enough number of initial keys. For example, when an auxiliary node preloads 3000 initial keys and the number of common auxiliary nodes increases from 2 to 3, the number of root keys required in ordinary node decreases by only 3.29% (from 152 to 147).

5.2.2. The Size of Key Group. Each initial key in the same key group can be verified by the root key. Hence, the larger the size of a key group, the more initial keys can the ordinary nodes verify. That is, the ordinary nodes require storing fewer root keys. Figure 4 shows how the size of key group influences the number of root keys stored in an ordinary node when different initial keys are preloaded in the auxiliary nodes.

It is worthy to point out that the curve will be smoother when the size of the key group increases, which means that the root keys decrease slowly. For example, when the number of preloaded initial keys increases from 1000 to 3000, the number of preloaded root keys will decrease 32.8% (from 368 to 247) when $M = 4$, and 13.8% (from 116 to 100) when $M = 16$. Hence, if the size of the key group is large enough, increasing the number of initial keys preloaded in auxiliary node cannot remarkably impact the network connectivity. It shows that the size of the key group is a critical factor that determines the number of preloaded root keys.

5.3. Comparison with Other Existing Schemes

5.3.1. Comparison with Basic Scheme. In the basic scheme [7], a sensor node must preload more keys with increased size of key pool to achieve desired network connectivity performance. Due to resource constraints in sensor nodes, the size of the key pool cannot be too large. On the other hand, a larger key pool size is desirable to prevent the adversary to capture more sensor nodes by exposing the key pool.

Figure 5 compares the storage overhead of the basic scheme and RPKE varying with the size of the key pool. It shows that not only does the number of root keys preloaded in ordinary node is smaller than the number of keys preloaded in node in the basic scheme, but also the number of preloaded root keys will hardly increase in the RPKE, no matter how the size of the key pool is selected if the number of key groups is fixed. Compared with the basic scheme, a significant characteristic in RPKE is that it can properly tune to the size of the key group so as to keep the almost same number of root keys required to store in the ordinary node no matter how large the size of the key pool is. It also proves that RPKE would be more efficient when the size of the key pool is relatively large and it is more suitable for large-scale networks.

5.3.2. Comparison with q-composite Scheme. The q-composite scheme improves the resilience against node compromise by increasing the threshold of common keys, thus more keys are preloaded in sensor or the size of key pool must be reduced.

Table 1: The setting of parameters in the simulation.

Parameters	Scenario 1	Scenario 2
Total Number of Sensor Nodes	1000	500
Field	1000 m × 1000 m	
Transmission Range of Ordinary Nodes	40 m	
Transmission Range of Auxiliary Nodes	40 m, 60 m, 80 m	
The Proportion of Auxiliary Nodes	10%, 15%	

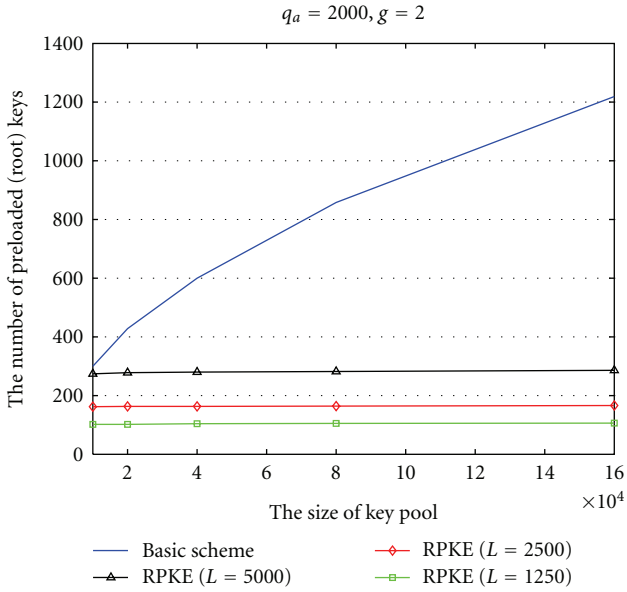


Figure 5: RPKE versus the Basic Scheme.

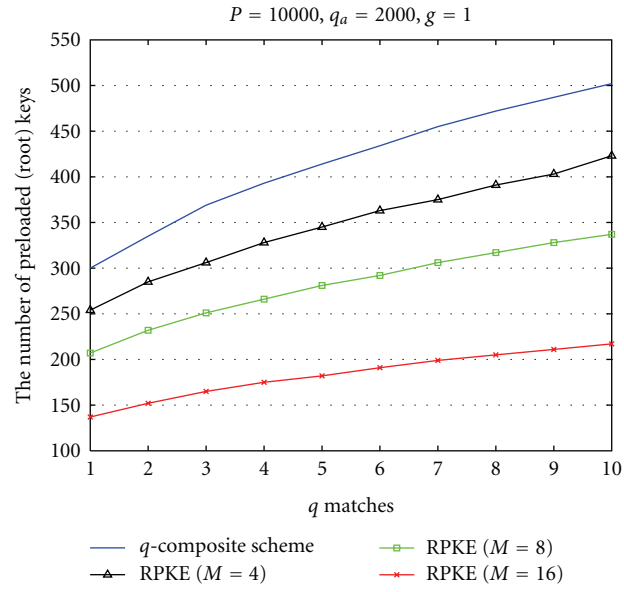


Figure 6: RPKE versus The q-composite Scheme.

Figure 6 compares q-composite scheme with RPKE with the size of key group. It can be seen that the total number of preloaded root keys in RPKE is smaller than that in the q-composite scheme. Moreover, when the size of key groups increases slightly, the number of preloaded root keys in ordinary nodes can reduce remarkably. Hence, RPKE increases only fewer root keys to achieve higher key match if the size of the key group is large enough.

5.4. Simulations

5.4.1. Hop Count. As aforesaid, the number of common auxiliary nodes between two neighboring ordinary nodes will affect the hop count h greatly. We set up two scenarios in which the main parameters used in simulation are shown in Table 1. Obviously, based on the above setting, the network degrees of two scenarios are 12 and 8, respectively. In the experiments, all the sensor nodes are uniformly distributed in the field, and each simulation result is averaged over 1,000 times.

Figures 7(a)–7(d) show the simulation results, where "Degree" is network degree and "Ratio" is the proportion of auxiliary nodes to the total nodes. "R" and "r" are the transmission ranges of auxiliary nodes and ordinary nodes, respectively. It can be easily observed that the larger the hop count or the higher network degree is, the more the number of common auxiliary nodes. Similarly, increasing the transmission of auxiliary nodes can also effectively increase the number of common auxiliary nodes. Hence, we have several different strategies to obtain a certain number of common auxiliary nodes according to the application.

5.4.2. Verification of Theoretical Result. To verify the theoretical results in (8), we design five scenarios in which different parameters are given under the 99.99% network connectivity probability. We consider a network of 100 nodes in which each node can reach to another node within 1-hop; moreover, any two neighboring ordinary nodes can obtain initial keys from the required number of common auxiliary nodes. The simulation results are shown in Figure 8.

In all five scenarios, the average simulation values are within 1.89% of those calculated with (8).

5.5. Overload Analysis. We will analyze the overheads for the RPKE in term of storage, communication, and computation cost, respectively. The auxiliary nodes are ignored since they do not participate in normal network operation.

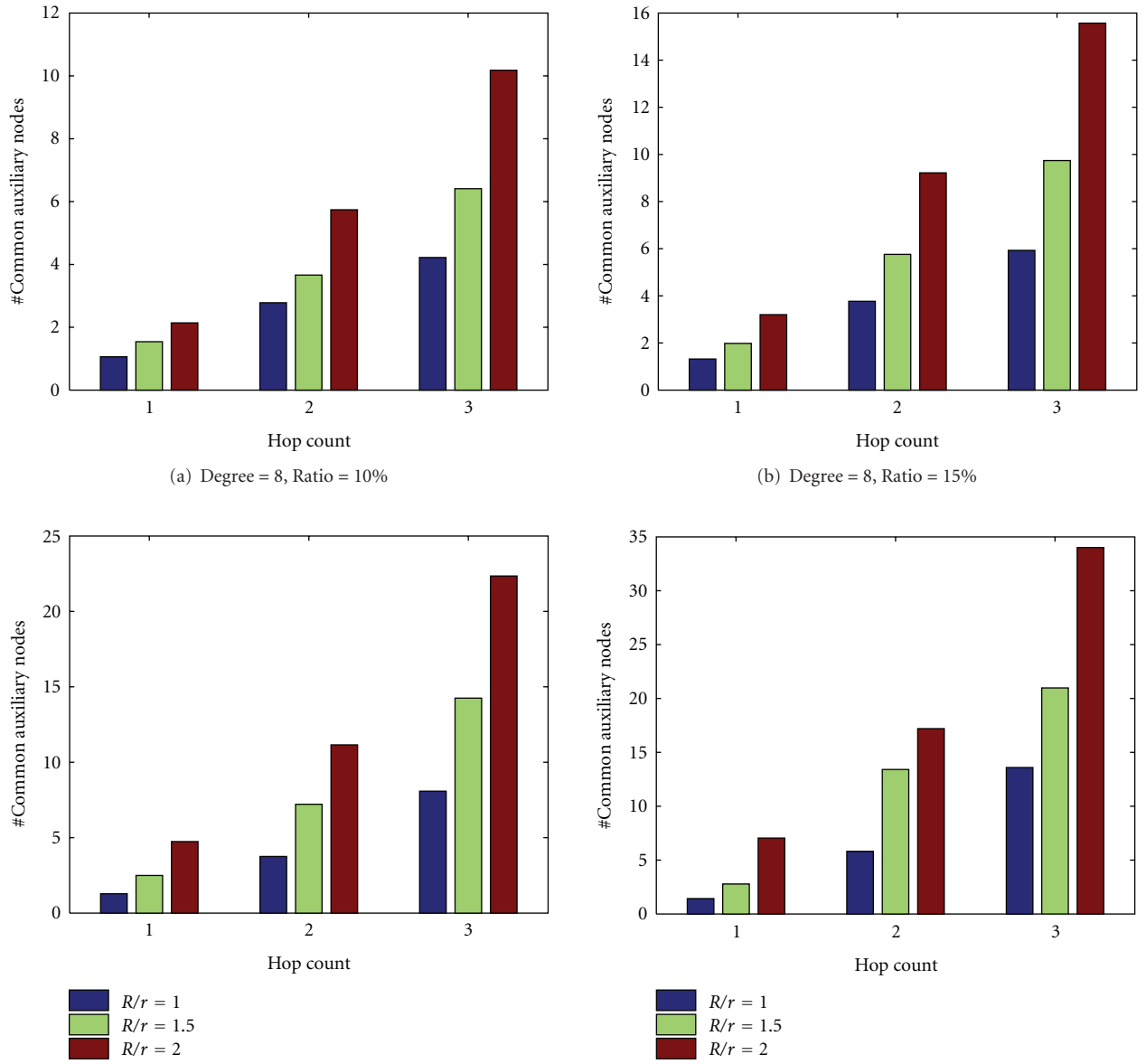


FIGURE 7: Hop count versus the number of common auxiliary nodes between two neighboring ordinary nodes varying with transmission range of auxiliary nodes, network degree, and the proportion of auxiliary nodes.

5.5.1. Storage Overload. The storage overhead of RPKE consists of the number of root keys and the key group identifiers held by an ordinary node. Clearly, as the aforesaid analysis in Section 5.1, to achieve the same network connectivity probability, RPKE requires fewer preloaded keying materials than other similar schemes. As shown in Figure 5, to achieve 99.99% network connectivity probability, ordinary node only needs to preload about 100 keys while the number of preloaded keys in basic scheme is about 1200 when the key pool is set to $16 * 10^4$ and the corresponding size of key group is 128. Even in such case, the storage requirement of RPKE is 200 keys if the key group identifier has the same length as the root key.

5.5.2. Communication Overload. According to the above discussions in Section 4.3, if two neighboring nodes do not share any root keys, they need to find a key path to establish their pairwise key. Now we analyze the required number of hops to establish a pairwise key and the communication overhead distributed on each hop through simulations. The network connectivity probability and the communication overhead weight varying with different network degree and the length path are shown in Figure 9.

The simulation results show that communication overhead of pairwise key establishment in the dense network is mainly distributed within first three hops. The smaller the maximum number of hops, the less the communication

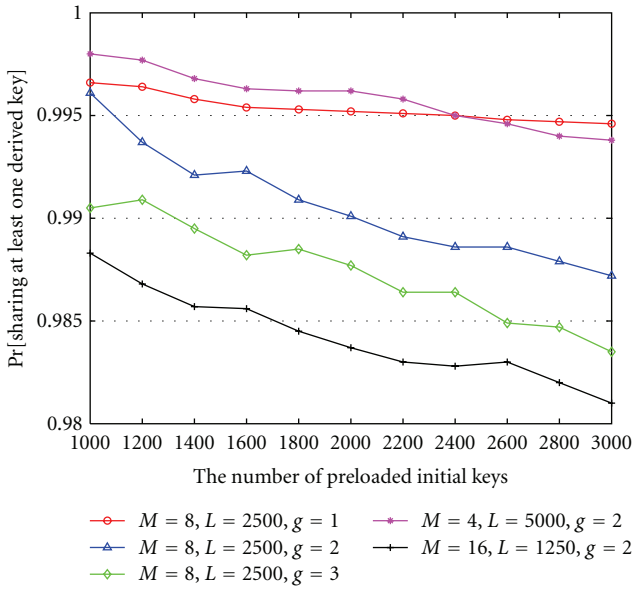


Figure 8: Simulation results under different system parameters.

overhead involved if the network connectivity probability is given.

5.5.3. Computation Overload. The computational overhead of an arbitrary ordinary node consists of two parts: *initial keys authentication* and *pairwise key establishment*. Ordinary node must authenticate each initial key sent from the common auxiliary nodes. To verify an initial key taking $(\log_2 M + 2)$ hash operations (M is the size of key group), it will take one hash operation to generate a derived key from each authenticated initial key. Since each pairwise key is constructed by m derived keys, hence, an ordinary node takes $m * (\log_2 M + 3)$ hash operations to generate a pairwise key. Assume the network degree is d, the computational overhead for an arbitrary ordinary node is $d * m * (\log_2 M + 2)$ hash operations.

Note that the computational overhead of iPAK [20], LKE [21], and SBK [22] is $O(\lambda)$ (λ is a security parameter which has the property that as long as no more than λ nodes are compromised, all communication links of noncompromised nodes remain secure) modular multiplications. Each modular multiplication takes 810 ms on the Atmega 128 L 8 M processor [25], while a RC5 hash operation takes only 5.6 ms [26]. Thus, RPKE is more scalable than iPAK, LKE, or SBK.

Different from the unbalanced random key predistribution scheme [18] as described in Section 2, in which L2 nodes must establish secure connections with the L1 nodes, in RPKE, the auxiliary nodes need not establish the pairwise key with the ordinary nodes or other auxiliary nodes. The ordinary nodes have less storage/computation/communication overload than L2 or L1 nodes.

5.6. Summary. The above theoretical and simulation analysis shows that RPKE significantly outperforms previous counterparts in terms of network connectivity, resilience against node compromise, and communication overheads. Such improvement is attributed to the role of auxiliary nodes and the property of key groups using Merkle hash tree, which efficiently enhance the correlation of initial keys preloaded in the auxiliary nodes and thus increase the chance for ordinary nodes to generate derived keys. Moreover, to achieve excellent network connectivity and high security strength, RPKE scheme can conveniently tune: (1) the size of key group, (2) the number of common auxiliary nodes, and (3) the total number of key groups.

6. Security Analysis

The security of pairwise key in RPKE relies on both the associated initial keys and root keys.

Theorem 3. *For two neighboring ordinary nodes U and V, if at least one of root key, or initial key, used to generate derived keys is secure, the pairwise key between them must be secure.*

Proof. According to (4), the derived key $K_{i,j}$ would be secure if the initial key $k_{i,j}$ or the root key R_i is secure (assume the identifiers of auxiliary nodes are public). It can also be concluded from (5) that the pairwise key will be secure if only at least one of its derived keys is secure. □

In the subsection, we discuss how the ordinary nodes or auxiliary nodes compromise impacts the network security.

6.1. Sensor Nodes Compromise. From the system's perspective, the adversary can capture any number of auxiliary nodes or ordinary nodes arbitrarily. So they have higher probability of compromising a fraction of auxiliary nodes and ordinary nodes in a special region. Now we discuss how the sensor nodes compromise affects the secure communication link between two noncompromised ordinary nodes, that is, the probability of pairwise key between two noncompromised ordinary nodes being compromised when there are a fraction of compromised sensor nodes.

We assume there are x compromised sensor nodes, in which the proportion of compromised auxiliary nodes is f_c. Hence, there are xf_c compromised auxiliary nodes and $x(1 - f_c)$ compromised ordinary nodes.

If an auxiliary node is compromised, the adversary can acquire all its q_a preloaded initial keys. Hence, the probability of an initial key not preloaded by a compromised auxiliary node is $1 - q_a/(L \times M)$, when there are xf_c compromised auxiliary nodes, the probability of an initial key still being secure is $(1 - q_a/(L \times M))^{xf_c}$. Similarly, if an ordinary node is compromised, the adversary can acquire all its q_n preloaded root keys. Hence, the probability of a root key being not preloaded by a compromised ordinary node is $1 - q_n/L$, when there are $x(1 - f_c)$ compromised ordinary nodes, the probability of a root key still being secure is $(1 - q_n/L)^{x(1-f_c)}$.

According to Theorem 3, if an initial key or a root key is secure, the derived key generated by them must be secure, in other words, the derived key is compromised if both the associated initial key and root key are insecure. Therefore, when there are xf_c compromised auxiliary nodes

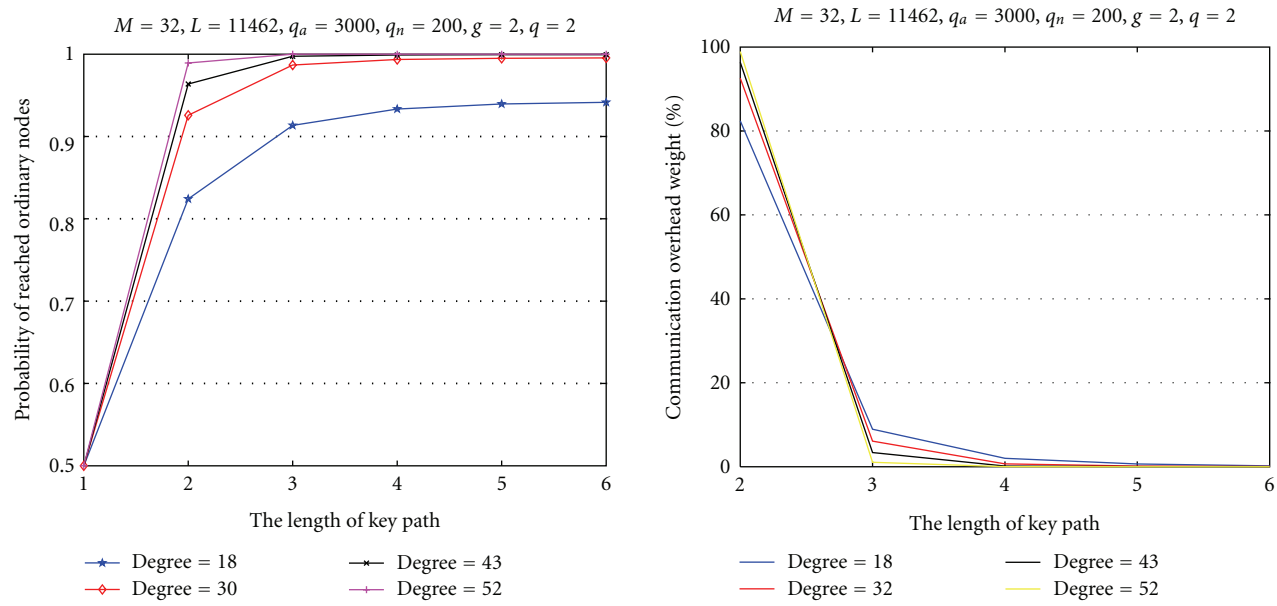


Figure 9: Communication overhead for different network degrees.

and $x(1-f_c)$ compromised ordinary nodes, the probability of a derived key being compromised is $(1-(1-q_n/L)^{x(1-f_c)})(1-(1-q_a/(L\times M))^{xf_c})$.

If the communication link key between two neighboring ordinary nodes is computed from m common derived keys, the probability of a communication link being compromised is$((1-(1-q_n/L)^{x(1-f_c)})(1-(1-q_a/(L\times M))^{xf_c}))^m$. Hence, the probability p_c that the communication links between two noncompromised neighboring ordinary nodes are compromised when there are xf_c compromised auxiliary nodes and $x(1-f_c)$ compromised ordinary nodes will be

$$p_c = \sum_{m=q}^{g\times q_a}\left(\left(1-\left(1-\frac{q_n}{L}\right)^{x(1-f_c)}\right)\left(1-\left(1-\frac{q_a}{L\times M}\right)^{xf_c}\right)\right)^m \times \frac{p(m)}{p_{\text{connect}}}. \tag{9}$$

Here, $p(m)$ and p_{connect} are defined as in (7) and (8), respectively, and g is the average number of common auxiliary nodes between two neighboring ordinary nodes.

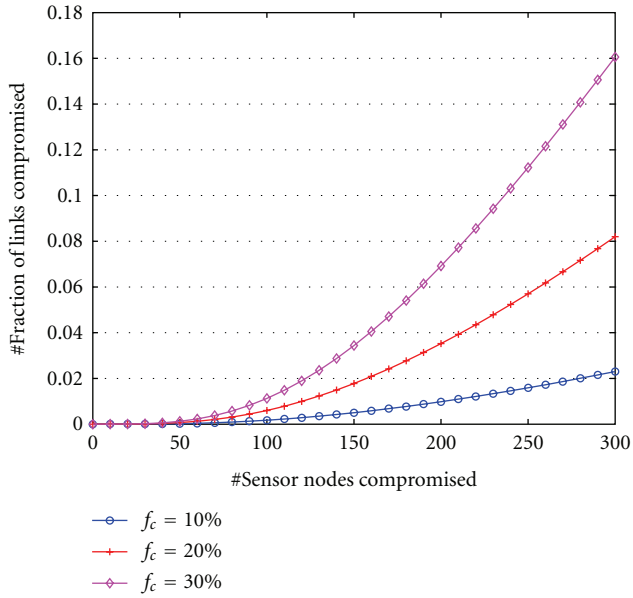
Figure 10 shows the resilience property of RPKE in the case of different network configuration. It is clear that p_c will increases with f_c for all scenarios. It will improve the resilience against node compromise with the increase of number of preloaded initial keys, or the size of key group, or the number of common auxiliary nodes.

The adversary may launch selective attacks, that is, he may select the auxiliary nodes to compromise if he can identify which node is auxiliary one. As a result, if the adversary can compromise all the auxiliary nodes during the pairwise keys establishment, the ordinary nodes may not establish pairwise keys with the help of their common auxiliary nodes. In practice, due to the similar appearance of the auxiliary nodes and ordinary nodes, there is no easy way to differentiate between them after the network deployment. The adversary would then randomly select an arbitrary number of auxiliary nodes or ordinary nodes to compromise. Hence, the value of f_c can be controlled within low bound.

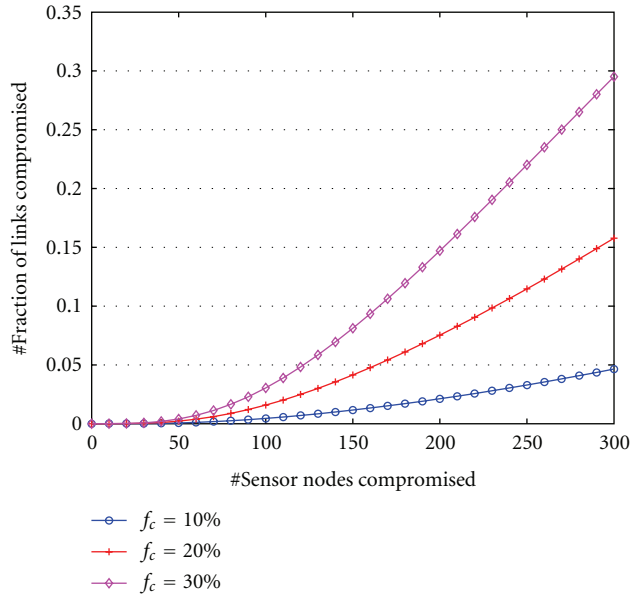
6.2. Comparison with Other Existing Schemes. We will compare the RPKE with the existing random key establishment schemes. Figure 11 shows the resilience property of the basic scheme [7], q-composite scheme [11], and the proposed RPKE.

Obviously, RPKE provides greater resilience against node compromise than the basic scheme or q-composite scheme. For example, when there are 250 compromised sensor nodes, the fraction of compromised communication links is 44.75% in the basic scheme and 48.27% in the q-composite scheme; whereas RPKE is only 5.73% (for $f_c = 10\%$), 19.09% (for $f_c = 20\%$), or 35.05% (for $f_c = 30\%$). Moreover, as shown in above analysis, we can tune some of the system parameters, such as the size of key group, the number of key group, or the number of auxiliary nodes, to improve the security strength in RPKE.

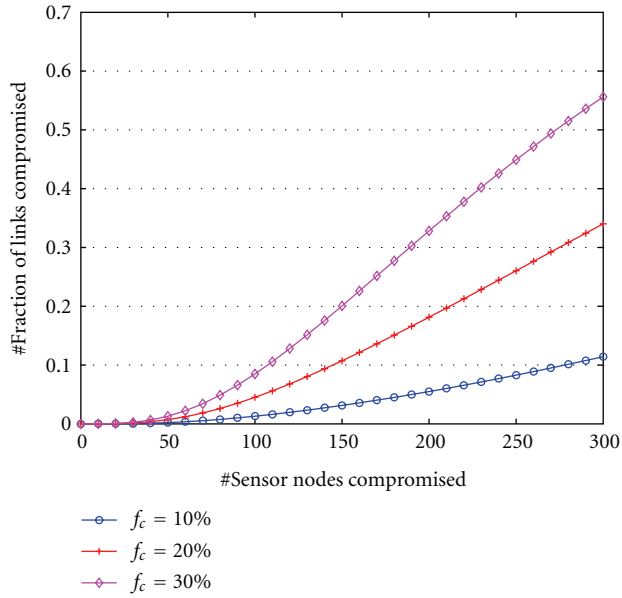
6.3. Other Attacks. The adversary may eavesdrop on all traffic or reply older packets. RPKE can defense such passive attacks efficiently. On one hand, since every packet sent by auxiliary nodes is encrypted by the hash image of root key, the adversary cannot obtain any secret information from such packet if it does not obtain the associated root key.



(a) $q_a = 3000, q_n = 200, M = 32, q = 2, g = 2$



(b) $q_a = 3000,\ q_n = 200,\ M = 32,\ q = 2, g = 1$



(c) $q_a = 2000,\ q_n = 200,\ M = 16,\ q = 2,\ g = 1$

Figure 10: Resilience against sensor nodes compromise, where $p_{\text{connect}} = 0.5$.

On the other hand, every packet broadcasted amongst ordinary nodes only includes the indices information; the adversary cannot acquire any secret information about the derived keys even though they can acquire these indices. According to the description in Section 4.2, adding the nonce in each packet can defense the reply attack.

The adversary may impersonate the common auxiliary nodes or use the compromised auxiliary nodes to fabricate initial keys. Such fake messages, however, cannot cause the ordinary nodes to accept the wrong initial keys because each ordinary node uses Merkle hash tree-based authentication method to verify the correctness of each received initial key.

The compromised ordinary nodes may send more request messages to other auxiliary nodes so as to compromise more initial keys. Some efficient mechanisms can be designed to defense such attack. For example, the distributed KDC is only in charge of disseminating initial keys to its neighboring ordinary nodes and replies those request messages which are consistent with what its two neighboring ordinary nodes claim. In other words, if two neighboring ordinary nodes claim different root key identifiers for the

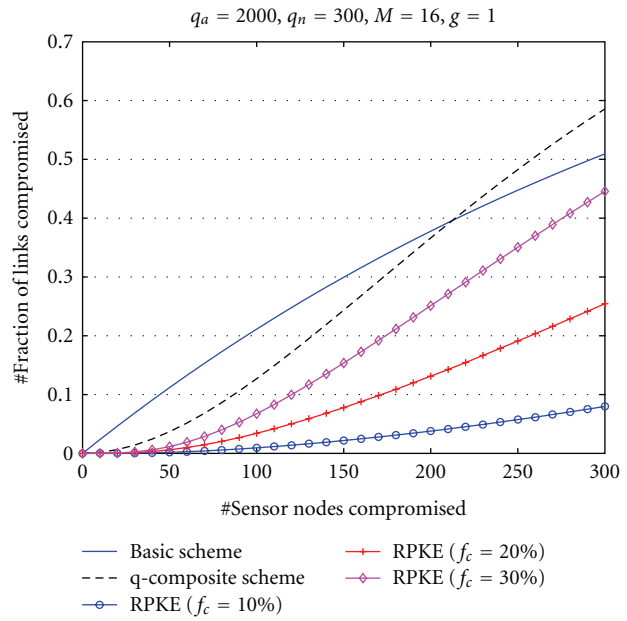


FIGURE 11: The fraction of compromised communication link between two noncompromised ordinary nodes in different schemes, where $p_{\text{connect}} = 0.5$.

same auxiliary node, the auxiliary node will refuse to reply such request messages.

Moreover, since the role of auxiliary nodes is to act as distributed KDC to help ordinary nodes establish pairwise keys, once pairwise key establishment procedure is finished, we can remove all the auxiliary nodes from the networks if we only consider static networks or there is no node addition during network operation.

7. Conclusions

In this paper, we have proposed a random pairwise key establishment scheme (RPKE) for WSNs. RPKE aims at achieving good performance in terms of network connectivity probability and strong security strength at the cost of low overheads for ordinary nodes. Theoretical and simulative evaluations demonstrate that it is an effective approach to use some additional nodes as distributed KDCs during the pairwise key establishment. RPKE not only reduces the storage requirement in ordinary nodes but also resist node compromise. Moreover, the system parameters in RPKE can be conveniently tuned to achieve excellent network connectivity and high security strength according to the applications.

Acknowledgments

This work is supported by the Natural Science Foundation of China (nos. 60932003, 61071065, and 60970101) and the National Grand Fundamental Research 973 Program of China (nos. 2010CB328105 and 2009CB320504).

References

[1] J. Kahn, R. Katz, and K. Pister, "Next century challenges: mobile networking for smart dust," in *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM '99)*, Seattle, Wash, USA, 1999.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[3] J. Chen, S. He, Y. Sun, P. Thulasiramanz, and X. Shen, "Optimal flow control for utility-lifetime tradeoff in wireless sensor networks," *Computer Networks*, vol. 53, no. 18, pp. 3031–3041, 2009.

[4] Crossbow Technology, "MICA2: Wireless measurement system," https://www.eol.ucar.edu/rtf/facilities/isa/internal/CrossBow/DataSheets/mica2.pdf.

[5] K. C. Koc, "High-speed RSA implementation," Tech. Rep., RSA Laboratories, 1994.

[6] B. C. Neuman and T. Tso, "Kerberos. An authentication service for computer networks," *IEEE Communications Magazine*, vol. 32, no. 9, pp. 33–38, 1994.

[7] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS '02)*, pp. 41–47, November 2002.

[8] Y. Jiang, C. Lin, M. Shi, and X. Shen, "Seal-healing group key distribution with time-limited user revocation for wireless sensor networks," *Ad Hoc Networks, Elsevier*, vol. 5, no. 1, pp. 14–23, 2007.

[9] Y. Jiang, C. Lin, X. Shen, and M. Shi, "A DoS and fault tolerant authentication protocol for group communications in Ad Hoc networks," *Computer Communications, Elsevier*, vol. 30, no. 1, pp. 2428–2441, 2007.

[10] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "SPINS: security protocols for sensor networks," *Wireless Networks*, vol. 8, no. 5, pp. 521–534, 2002.

[11] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *Proceedings of the IEEE Symposium on Security And Privacy*, pp. 197–213, May 2003.

[12] S. Blackshear and R. M. Verma, "R-LEAP+: randomizing LEAP+ key distribution to resist replay and jamming attacks," in *Proceedings of the ACM Symposium on Applied Computing*, pp. 1985–1992, 2010.

[13] D. Liu and P. Ning, "Establishing pairwise keys in distributed sensor networks," in *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS '03)*, pp. 52–61, October 2003.

[14] W. Du, Y. S. Han, J. Deng, and P. K. Varshney, "A pair-wise key pre-distribution scheme for wireless sensor networks," in *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS '03)*, pp. 42–51, October 2003.

[15] D. Liu and P. Ning, "Location-based pair-wise key establishments for static sensor networks," in *Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor networks*, pp. 72–82, October 2003.

[16] D. Huang, M. Mehta, D. Medhi, and L. Harn, "Location-aware key management scheme for wireless sensor networks," in *Proceedings of the ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '04)*, pp. 29–42, October 2004.

[17] W. Du, J. Deng, Y. S. Han, S. Chen, and P. K. Varshney, "A key management scheme for wireless sensor networks using deployment knowledge," in *Proceedings of the 23rd Annual*

Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04), pp. 586–597, Hong Kong, March 2004.

[18] P. Traynor, H. Choi, G. Cao, S. Zhu, and T. La Porta, "Establishing pair-wise keys in heterogeneous sensor networks," in *Proceedings of the 25th Annual Conference of the IEEE Computer and Communications Societies (INFOCOM '06)*, Barcelona, Spain, April 2006.

[19] T. Vu, R. Safavi-Naini, and C. Williamson, "Securing wireless sensor networks against large-scale node capture attacks," in *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security (ASIACCS '10)*, pp. 112–123, Beijing, China, April 2010.

[20] L. Ma, X. Cheng, F. Liu, F. An, and J. Rivera, "iPAK: an in-situ pair-wise key bootstrapping scheme for wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 8, pp. 1174–1184, 2007.

[21] F. Liu and X. Cheng, "LKE: a self-configuring scheme for location-aware key establishment in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 7, no. 1, pp. 224–232, 2008.

[22] F. Liu, X. Cheng, L. Ma, and K. Xing, "SBK: a self-configuring framework for bootstrapping keys in sensor networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 7, pp. 858–868, 2008.

[23] R. C. Merkle, "Protocols for public key cryptosystems," in *Proceedings of the IEEE Symposium on Research in Security and Privacy*, p. 122, 1980.

[24] R. Rivest, "The RC5 encryption algorithm," in *Proceedings of the 1st International Workshop on Fast Software Encryption*, vol. 809, pp. 86–96, Leuven Belgium, December 1994.

[25] A. S. Wandert, N. Gura, H. Eberle, V. Gupta, and S. C. Shantz, "Energy analysis of public-key cryptography for wireless sensor networks," in *Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications (PerCom '05) 2005*, pp. 324–328, March 2005.

[26] P. Ganesan, R. Venugopalan, P. Peddabachagari, A. Dean, F. Mueller, and M. Sichitiu, "Analyzing and modeling encryption overhead for sensor network nodes," in *Proceedings of the 2nd ACM International Workshop on Wireless Sensor Networks and Applications (WSNA '03)*, pp. 151–159, September 2003.